

IN THE CLAIMS:

The following listing of the claims replaces all prior claim listings and versions:

1-10. (Cancelled)

11. (Previously presented) A method in a LSI design and development process for evaluating an architecture design for an algorithm design by performing a performance evaluation of at least one bus at a high-level stage of said design and development process, said method comprising:

structuring source code describing the algorithm design in a general purpose high-level programming language by isolating elements of said source code representing hardware units and software units;

creating an evaluation function for evaluating data transfer that occurs on said at least one bus, the bus being a part of the source code realizing the data transfer between said elements representing hardware units and software units;

modifying at least one element of said source code elements based on a result of an implementation of said evaluation function; and

performing said performance evaluation by simulating said modified source code elements and evaluating said data transfer on the bus.

12. (Currently amended) The method according to claim [[1]] 11, further comprising:

restructuring the source code based on [[o]] the evaluated data transfer; and

performing said performance evaluation again by simulating said restructured source code again.

13. (Previously presented) The method according to claim 11, wherein a bus traffic is calculated from the evaluated data transfer with respect to the processing rate of the bus.

14. (Previously presented) The method according to claim 11, further comprising:

feeding back a result of the performance evaluation of the bus to the step of structuring the source code to improve the architecture design at a high-level design stage by isolating in said source code new elements representing hardware units and new elements representing software units.

15. (Currently amended) The method according to claim 14, wherein in response to the bus traffic, isolation of the source code into~~in~~ elements representing hardware units and elements representing software units is optimized.

16. (Currently amended) The method according to claim 11, further comprising:
after creating the evaluation function, sequentially reading in the source code line by line while effecting syntax analysis;

determining whether the source code is to be modified based on whether a line of source code represents writing data to variables that are defined in advance and are loaded onto the bus to be evaluated;

upon determining that the source code is to be modified,
modifying the source code by embedding the evaluation function one of immediately before or immediately after the line of source code in which the variable is written;
repeating the forgoing steps until the source code is completely read in up to a last line of source code;

structuring the source code ~~into~~ⁱⁿ elements representing at least one of the hardware units and the software units for use in the architecture design by compiling said structured source code elements;

calculating the data transfer rate on the bus by executing the compiled source code elements ~~in~~^{at} a simulation program;

calculating bus traffic with regard to a given processing rate of the bus; and

performing evaluation of the performance of the bus in response to the bus traffic.

17. (Previously presented) The method according to claim 16, wherein the variables loaded onto the bus consist of n bits while the bus consists of m bit lines, where n and m are both integers, and n is a multiple of m , and the bus traffic for the processing rate is produced such that the number of times in effecting data transfer on the bus is multiplied by n/m and is then divided by the processing rate.

18. (Previously presented) The method according to any one of the preceding claims, wherein the general purpose high-level programming language is one of C language and C++ language.

19. (Previously presented) The method according to claim 11,
wherein the evaluation function increments a counting value if a pre-defined variable is loaded onto the bus.